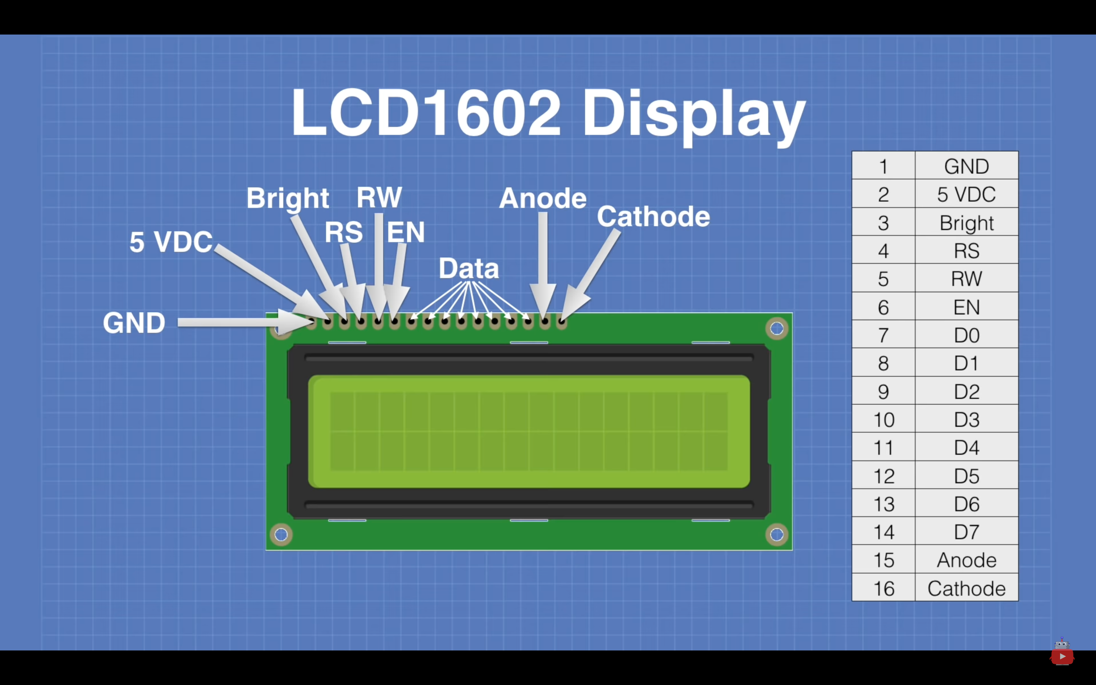
Arduino project

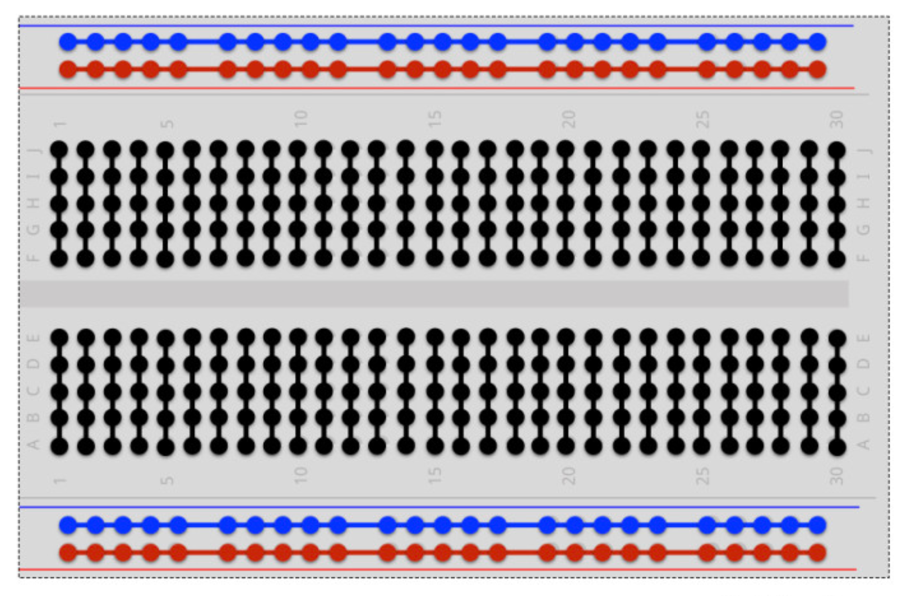
1. **LCD:**



* A **register select (RS) pin** that controls where in the LCD's memory you're writing data to. You can select either the data register, which holds what goes on the screen, or an instruction register, which is where the LCD's controller looks for instructions on what to do next.
* A **Read/Write (R/W) pin** that selects reading mode or writing mode
* An **Enable pin** that enables writing to the registers
* 8 **data pins (D0 -D7)**. The states of these pins (high or low) are the bits that you're writing to a register when you write, or the values you're reading when you read.

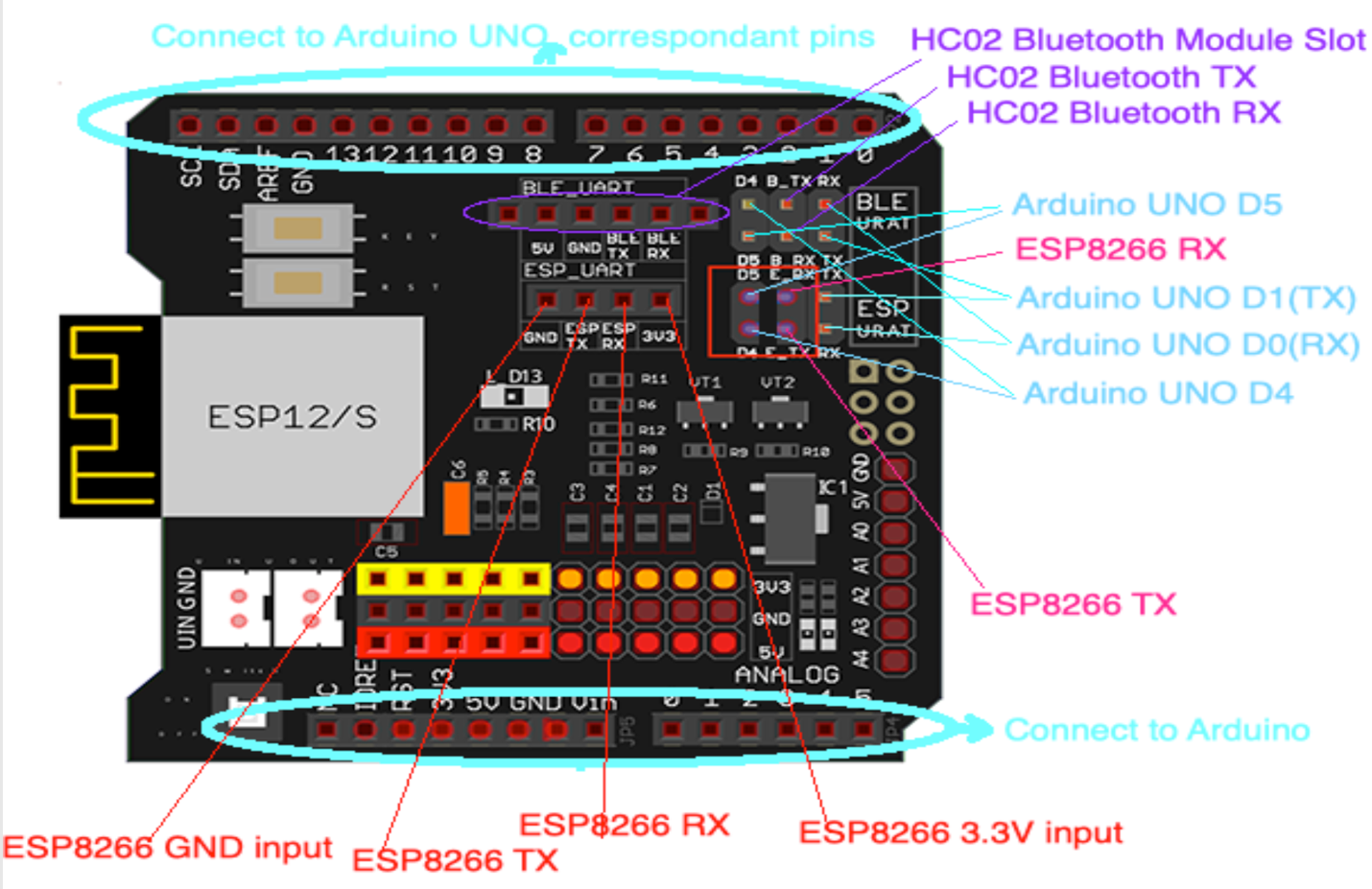
**Start connecting the Arduino with the LCD**

https://docs.arduino.cc/learn/electronics/lcd-displays



* Have to use LiquidCrystal\_I2C library, install it using Sketch -> include library -> manage library -> find LiquidCrystal\_I2C.
* Then use example code -> “hello world”.

1. **Connect with wifi shield**



**CODE:**

#include <CytronWiFiShield.h>

#include <CytronWiFiClient.h>

#include <CytronWiFiServer.h>

#include <SoftwareSerial.h>

#include <Wire.h>

#include <LiquidCrystal\_I2C.h>

LiquidCrystal\_I2C lcd(0x27,20,4);

const char \*ssid = "Schok Volt SV55 9223";

const char \*pass = "Rinvo0368";

ESP8266Server server(80);

String USDebt;

bool dollar = false;

const char htmlHeader[] = "HTTP/1.1 200 OK\r\n"

"Content-Type: text/html\r\n"

"Connection: close\r\n\r\n"

"<!DOCTYPE HTML>\r\n"

"<html>\r\n";

void setup() {

// put your setup code here, to run once:

Serial.begin(9600);

while (!Serial) {

; // wait for serial port to connect. Needed for Leonardo only

}

if(!wifi.begin(4, 5))

{

Serial.println(F("Error talking to shield"));

while(1);

}

Serial.println(wifi.firmwareVersion());

Serial.print(F("Mode: "));Serial.println(wifi.getMode());// 1- station mode, 2- softap mode, 3- both

// Uncomment these 2 lines if you are using static IP Address

// Serial.println(F("Setup wifi config"));

// wifi.config(ip);

Serial.println(F("Start wifi connection"));

if(!wifi.connectAP(ssid, pass))

{

Serial.println(F("Error connecting to WiFi"));

while(1);

}

Serial.print(F("Connected to "));Serial.println(wifi.SSID());

clientTest();

server.begin();

lcd.init(); // initialize the lcd

lcd.init();

// Print a message to the LCD.

lcd.backlight();

lcd.setCursor(3,0);

lcd.print("Hello, world!");

lcd.setCursor(2,1);

lcd.print("Ywrobot Arduino!");

}

void loop() {

// put your main code here, to run repeatedly:

serverTest();

}

void serverTest()

{

ESP8266Client client = server.available();

if(client.available()>0)

{

String req = client.readStringUntil('\r');

// First line of HTTP request looks like "GET /path HTTP/1.1"

// Retrieve the "/path" part by finding the spaces

int addr\_start = req.indexOf(' ');

int addr\_end = req.indexOf(' ', addr\_start + 1);

if (addr\_start == -1 || addr\_end == -1) {

Serial.print(F("Invalid request: "));

Serial.println(req);

return;

}

req = req.substring(addr\_start + 1, addr\_end);

Serial.print(F("Request: "));

Serial.println(req);

client.flush();

if(req.equals("/"))

{

IPAddress ip = wifi.localIP();

String ipStr = String(ip[0]) + '.' + String(ip[1]) + '.' + String(ip[2]) + '.' + String(ip[3]);

client.print(htmlHeader);

String htmlBody = "Hello from ESP8266 at ";

htmlBody += ipStr;

htmlBody += "</html>\r\n\r\n";

client.print(htmlBody);

}

else if(req.equals("/analog"))

{

client.print(htmlHeader);

String htmlBody="";

for (int a = 0; a < 6; a++)

{

htmlBody += "A";

htmlBody += String(a);

htmlBody += ": ";

htmlBody += String(analogRead(a));

htmlBody += "<br>\r\n";

}

htmlBody += "\r\n</html>\r\n";

client.print(htmlBody);

}

else if(req.equals("/info"))

{

String toSend = wifi.firmwareVersion();

toSend.replace("\r\n","<br>");

client.print(htmlHeader);

client.print(toSend);

client.print("</html>\r\n");

}

else

client.print("HTTP/1.1 404 Not Found\r\n\r\n");

client.stop();

}

}

void clientTest()

{

const char destServer[] = "164.92.67.221";

ESP8266Client client;

if (!client.connect(destServer, 80))

{

Serial.println(F("Failed to connect to server."));

client.stop();

return;

}

const char \*httpRequest = "GET /api/national\_debt HTTP/1.1\r\n"

"Host: 164.92.67.221\r\n"

"Connection: close\r\n\r\n";

if(!client.print(httpRequest))

{

Serial.println(F("Sending failed"));

client.stop();

return;

}

// set timeout approximately 5s for server reply

int i=5000;

while (client.available()<=0&&i--)

{

delay(1);

if(i==1) {

Serial.println(F("Timeout"));

return;

}

}

while (client.available()>0)

{

char c = (char)client.read();

//Serial.print(c);

if(c == '$'){

dollar = true;

}

if(dollar == true){

if(isdigit(c)){

Serial.write(c);

}

}

}

client.stop();

}

Mathlab:

% Enter your MATLAB Code below

SDWeather = webread('https://forecast.weather.gov/MapClick.php?lat=32.92926&lon=-117.13108&FcstType=json');

date = str2double(SDWeather.currentobservation.Date)

temp = str2double(SDWeather.currentobservation.Temp)

humidity = str2double(SDWeather.currentobservation.Relh)

wind = str2double(SDWeather.currentobservation.Winds)

<https://forecast.weather.gov/MapClick.php?lat=32.7157&lon=-117.1617>

my API key on openweathermap.org:

9714d1a24672166514aeafd864615b95

https://api.openweathermap.org/data/2.5/weather?lat=32.7157&lon=-117.1617&appid= 9714d1a24672166514aeafd864615b95

check if the api key work:

api.openweathermap.org/data/2.5/weather?q=san diego,US&appid=9714d1a24672166514aeafd864615b95